In the Claims

1.   (Currently amended) A method of processing a database service query, comprising:

receiving a service query, ~~including~~ the service query comprising a filter ~~having~~ that comprises one or more filter items;

expanding the filter of the service query; and

applying a condition test to each filter item ~~to determine~~ of the filter, the condition test comprising:

determining if the filter item ~~includes~~ comprises a NOT ~~connective~~ connective; and

determining if the filter item ~~is~~ comprises one of ~~two types of filter items~~ a type only filter item or a type and value filter item.

2.   (Canceled)

3.   (Currently amended) The method as claimed in ~~claim 2~~ claim 1, ~~wherein~~ comprising applying a logical methodology ~~is applied~~ to evaluate NOT connectives associated with type only filter items.

4.   (Original) The method as claimed in claim 3, wherein the logical methodology comprises a subtraction method.

5.   (Currently amended) The method as claimed in ~~claim 2~~ claim 1, ~~wherein a~~ further comprising, if it is determined that the filter item comprises a NOT connective and a type and value filter item, pushing the NOT connective associated ~~with a~~ with the type and value filter item ~~is pushed~~ inside the filter item, resulting in changing an operator inside the filter item.

6.   (Currently amended) The method as claimed in claim 1, wherein the condition test further ~~includes~~ comprises determining if each filter item can be pre-evaluated to true.

7.   (Currently amended) The method as claimed in claim 1, wherein the condition test further ~~includes~~ comprises determining if each filter item can be pre-evaluated to false, such that ~~the~~ an expanded term of the expanded filter can be ignored.

8.   (Currently amended) The method as claimed in claim 1, wherein expanding the filter comprises expanding the filter ~~is expanded~~ to a minimum set of terms.

9.   (Currently amended) A directory services arrangement comprising:

a plurality of tables, each table ~~having~~ comprising a plurality of rows and columns ~~storing,~~ operable to store arbitrary data in a search service, ~~wherein~~ at least one of the tables ~~is has~~ comprising information ~~used to resolve~~ for resolving filters ~~having~~ that comprise at least one filter item;

means for expanding each filter into an expanded term; and

condition test means ~~for determining~~ operable to determine:

whether each filter item ~~includes~~ comprises a NOT ~~connective~~ connective; and

whether each filter item ~~is~~ comprises one of ~~two types of filter items~~ a type only filter item or a type and value filter item.

10.   (Canceled)

11.   (Currently amended) The directory services arrangement ~~method~~ as claimed in ~~claim 10~~ claim 9, wherein a logical methodology is applied to evaluate NOT connectives associated with type only filter items.

12.   (Original) The directory services arrangement as claimed in claim 11, wherein the logical methodology comprises a subtraction method.

13.   (Currently amended) The directory services arrangement as claimed in ~~claim 10~~ claim 9, wherein, if it is determined that the filter item comprises a NOT connective and a type and value filter item, a NOT connective associated with a type and value filter item is pushed inside the filter item resulting in changing an operator inside the filter item.

14.   (Currently amended) The directory services arrangement as claimed in claim 9, wherein the condition test means is further ~~determines~~ operable to determine if each filter item can be pre-evaluated to be true.

15.   (Currently amended) The directory services arrangement as claimed in claim 9, wherein the condition test means is further ~~determines~~ operable to determine if each filter item can be pre-evaluated to be false, such that ~~the~~ an expanded term of the expanded filter can be ignored.

16.   (Currently amended) The directory services arrangement as claimed in claim 9, wherein the condition test means is further ~~determines~~ operable to determine if each filter can be pre-evaluated to be true, but is inverted by a NOT connective, such that the expanded term can be ignored.

17.   (Currently amended) A method of processing a database service query, comprising:
    receiving a service query;
    applying a filter to the service query resulting in zero or more filter items; and
    applying, if one or more filter items results, a condition test to each filter item to determine ~~a form of~~ whether the filter item comprises one of a type only filter item or a type and value filter item.

18.   (Canceled)

19.   (Currently amended) The method as claimed in ~~claim 18~~ claim 17, further comprising evaluating the filter item in accordance with a logical methodology if the filter item is type only form.

20.   (Original) The method as claimed in claim 19, wherein the logical methodology comprises a subtraction method.

21.   (Currently amended) The method as claimed in claim 20, wherein the subtraction method ~~includes~~ comprises ~~the use of~~ using an ANSI SQL "except" clause.

22.   (Currently amended) The method as claimed in claim 20, wherein the subtraction method ~~transforms~~ comprises transforming each filter item to a form that contains fewer or no NOT connectives.

23.   (Canceled)

24.   (Currently amended) The method as claimed in ~~claim 23~~ claim 17, ~~wherein~~ comprising adding, if the filter item is a type and value form, ~~adding~~ SQL representing the filter item to an expression to be evaluated, which may involve at least one table join.

25.   (Currently amended) The method as claimed in ~~claim 23~~ claim 17, ~~wherein~~ comprising, if the filter item is an inverse of the type and value filter item, pushing the NOT connective inside the filter item.

26.   (Original) The method as claimed in claim 25, further comprising applying the pushed NOT connective to an operator.

27.   (Original) The method as claimed in claim 26, wherein the step of applying the pushed NOT is effected by inverting the operator.

28.   (Currently amended) A directory services arrangement comprising:

a plurality of tables, each table ~~having~~ comprising a plurality of rows and columns, ~~and storing~~ operable to store arbitrary data, ~~wherein~~ at least one of the tables ~~has~~ comprising information ~~used to resolve~~ for resolving filters that comprise at least one filter item in a search service; and

a condition tester ~~that determines~~ operable to determine:

whether each filter item comprises a NOT connective; and

whether ~~a filter~~ each filter item ~~is~~ comprises a type only filter item or a type and value filter item.

29.   (Canceled)

30.   (New) The directory services arrangement as claimed in claim 28, wherein a logical methodology is applied to evaluate NOT connectives associated with type only filter items.

31.   (New) The directory services arrangement as claimed in claim 30, wherein the logical methodology comprises a subtraction method.

32.   (New) The directory services arrangement as claimed in claim 28, wherein, if it is determined that the filter item comprises a NOT connective and a type and value filter item, a NOT connective associated with a type and value filter item is pushed inside the filter item resulting in changing an operator inside the filter item.

33.   (New) The directory services arrangement as claimed in claim 28, wherein the condition tester is further operable to determine if each filter item can be pre-evaluated to be true.

34.   (New) The directory services arrangement as claimed in claim 28, wherein the condition tester is further operable to determine if each filter item can be pre-evaluated to be false, such that an expanded term of the expanded filter can be ignored.

35.   (New) The directory services arrangement as claimed in claim 28, wherein the condition tester is further operable to determine if each filter can be pre-evaluated to be true, but is inverted by a NOT connective, such that the expanded term can be ignored.

36.   (New) Software for processing a database service query, the software being embodied in a computer-readable and when executed operable to:

receive a service query, the service query comprising a filter comprising one or more filter items;

expand the filter;

apply a condition test to each filter item of the filter, the condition test comprising:

determining if the filter item comprises a NOT connective; and

determining if the filter item comprises one of a type only filter item or a type and value filter item.